CSE 559A: Computer Vision

[credit: danjodon.deviantart.com]

Fall 2017: T-R: 11:30-1pm @ Lopata 101

Instructor: Ayan Chakrabarti (ayan@wustl.edu).
Staff: Abby Stylianou (abby@wustl.edu), Jarett Gross (jarett@wustl.edu)

http://www.cse.wustl.edu/~ayan/courses/cse559a/

Sep 26, 2017

GENERAL

- PSET 1 Due 11:59pm today.

- PSET 2 will be released by the end of today.
    - PSET 2 has one dependency on Prob 6 of PSET 1.
    - You will need the `im2wv` and `wv2im` functions.
    - If you couldn't get Prob 6 to work:
        - (Preferred) We'll post code after everyone's PSET 1 solutions are in.
        - If you've submitted your PSET 1 and rather not wait, post a private note on Piazza.

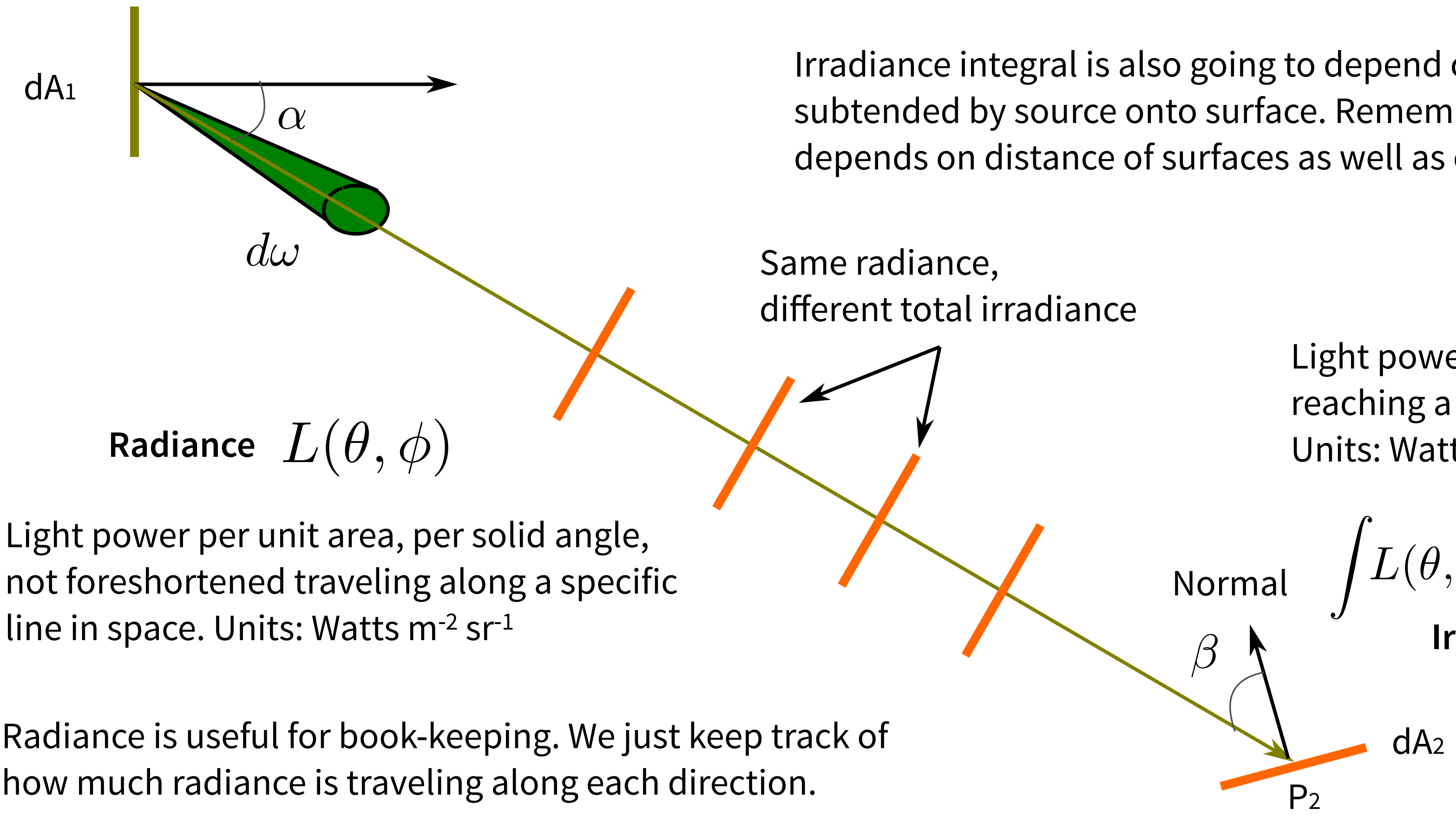
- Monday Office Hours Location: May change, will announce if it does.

RECAP

Radiance & Irradiance

Intensity = Energy = Power x Time = Irradiance across sensor x Time
= Will be proportional to Radiance

Irradiance integral is also going to depend on angle subtended by source onto surface. Remember, this depends on distance of surfaces as well as orientation.



Same radiance,
different total irradiance

Light power per unit area reaching a surface.
Units: Watts m^{-2}

Radiance $L(\theta, \phi)$

$$\int L(\theta, \phi) \cos \beta d\omega$$

Irradiance

Light power per unit area, per solid angle, not foreshortened traveling along a specific line in space. Units: Watts m^{-2} sr^{-1}

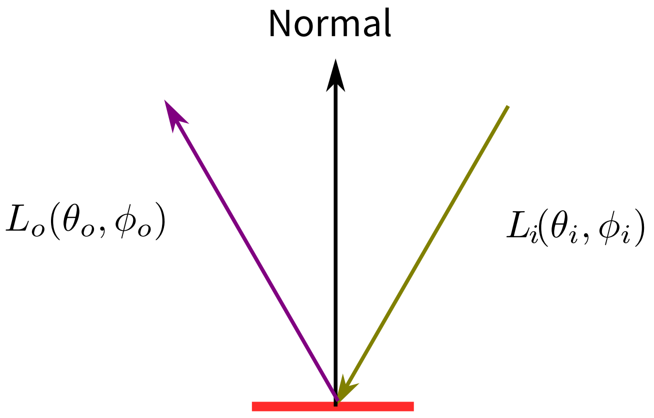
Radiance is useful for book-keeping. We just keep track of how much radiance is traveling along each direction.

Often we choose co-ordinate system where $\beta = \theta$

RECAP

Bi-directional Reflectance Distribution Function (BRDF)

- Radiance is Radiance, whether from a light source, or a reflected surface.



In co-ordinate system around surface normal

$$\rho(\theta_i, \phi_i, \theta_o, \phi_o) = \frac{L_o(\theta_o, \phi_o)}{L_i(\theta_i, \phi_i)\cos\theta_i d\omega_i}$$

Can be thought of as the ratio between output radiance,
due to incoming light from θ_i, ϕ_i
and input (infinitesimal) irradiance
at the surface from θ_i, ϕ_i

$$L_o(\theta_o, \phi_o) = \int p(\theta_i, \phi_i, \theta_o, \phi_o) L_i(\theta_i, \phi_i)\cos\theta_i d\omega_i$$

RECAP

Bi-directional Reflectance Distribution Function (BRDF)

Lambertian Surfaces

$$\rho(\theta_i, \phi_i, \theta_o, \phi_o) = \mathsf{K}$$

$$L_o(\theta_o, \phi_o) = \mathsf{K} \int L_i(\theta_i, \phi_i) \cos\theta_i d\omega_i$$

Output Radiance Proportional to Total Irradiance
Observed intensity depends on surface normal, lighting, but not viewing direction.

General Surfaces

Intensity Depends on Viewing Direction
(actually, even on aperture size)

RECAP

- We dropped color everywhere, but everything is also function of wavelength λ
  - Radiance, Irradiance, BRDF
- Fact about BRDFs: They are notoriously hard to measure.
- Common low-parameter models:
  - Lambertian
  - Some combination of lambertian + specular (e.g., Phong)
  - But many models more useful for Graphics than Vision
- Radiance is Radiance: Objects can act as light sources !
  - Inter-reflections
- We'll often deal with summations instead of integrals
  - But it is important to keep the derivations in mind.
  - Otherwise you forget to take into account stretching, foreshortening
- Additional Reference: Forsyth & Ponce: Chapters 4 & 5
- Less Detailed / Quick: Szeliski Sec 2.2

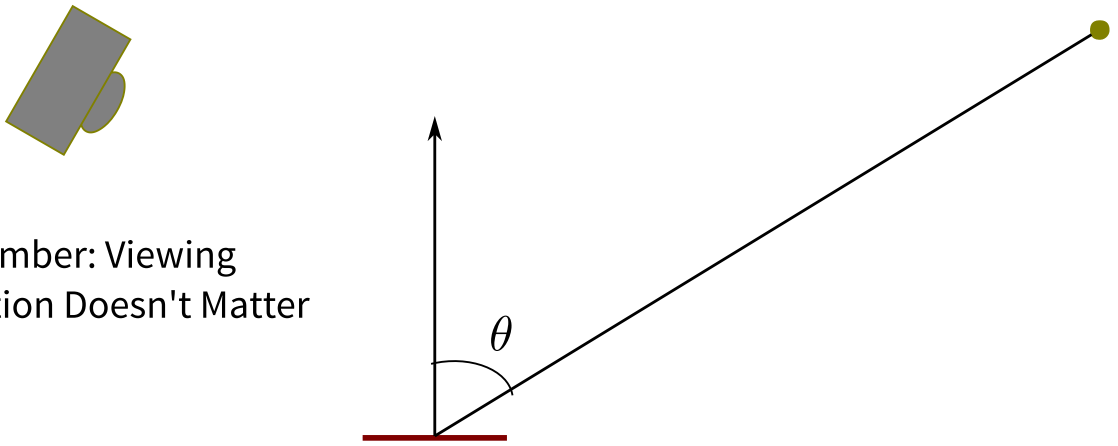
LIGHTS

Point Source

For Lambertian surface, reflected radiance in all directions, and therefore observed intensity:

$$I = \rho \ell \cos\theta$$

ρ proportional to constant BRDF / "albedo"
Often itself called albedo.



Remember: Viewing Direction Doesn't Matter

$$\int L_i(\theta, \phi) \cos\theta d\omega = \ell \cos\theta$$

Concentrates all its radiance in infinitely small subtended solid angle

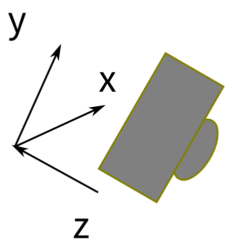
Spherical Light Source with radius → 0

LIGHTS

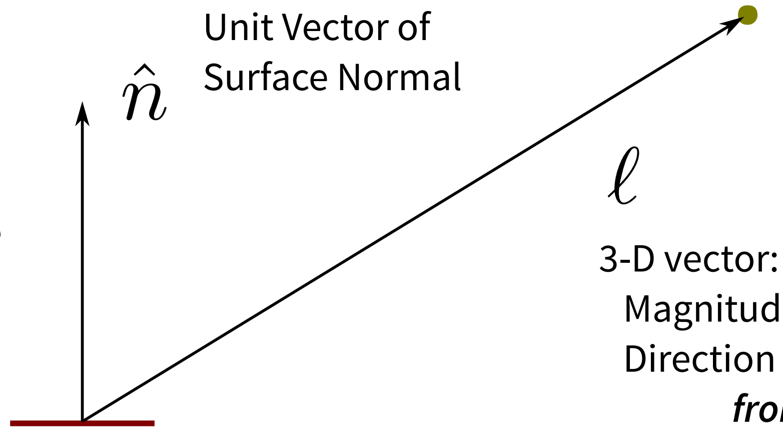
Point Source

For Lambertian surface, reflected radiance in all directions, and therefore observed intensity:

$$I = \rho \ell \cos \theta$$



Typically, choose co-ordinate system flipped along camera viewing direction.



3-D vector:
Magnitude is "intensity" of light
Direction is direction ***to*** light source ***from*** surface point.

$$I = \rho \langle \hat{n}, \ell \rangle$$

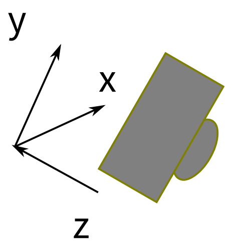
Vector Notation

LIGHTS

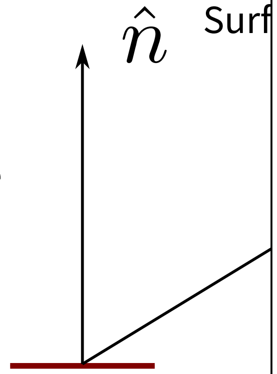
Point Source

For Lambertian surface, reflected radiance in all directions, and therefore observed intensity:

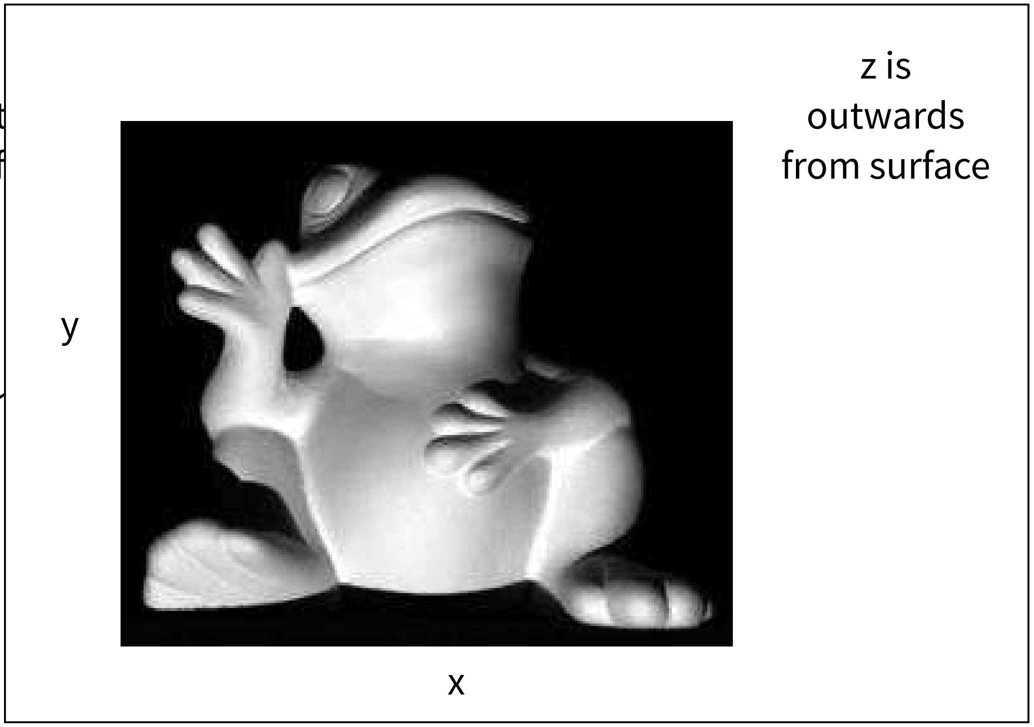
$$I = \rho \ell \cos\theta$$



Typically, choose co-ordinate system flipped along camera viewing direction.



Unit
Surf



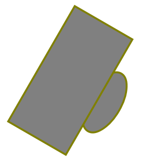
$$I = \rho\langle \hat{n}, \ell \rangle$$

Vector Notation

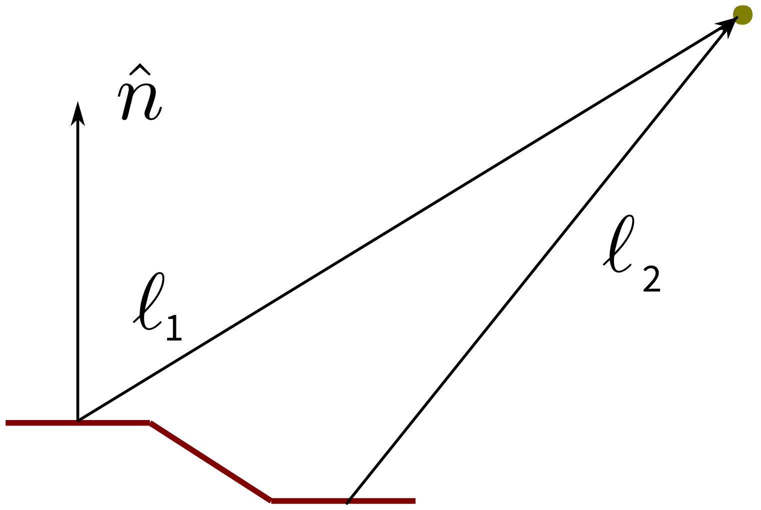
LIGHTS

Point Source

For Lambertian surface, reflected radiance in all directions, and therefore observed intensity:



$$I = \rho\langle \hat{n}, \ell \rangle$$

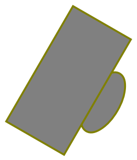


Lighting Direction / Vector will depend on Surface Location

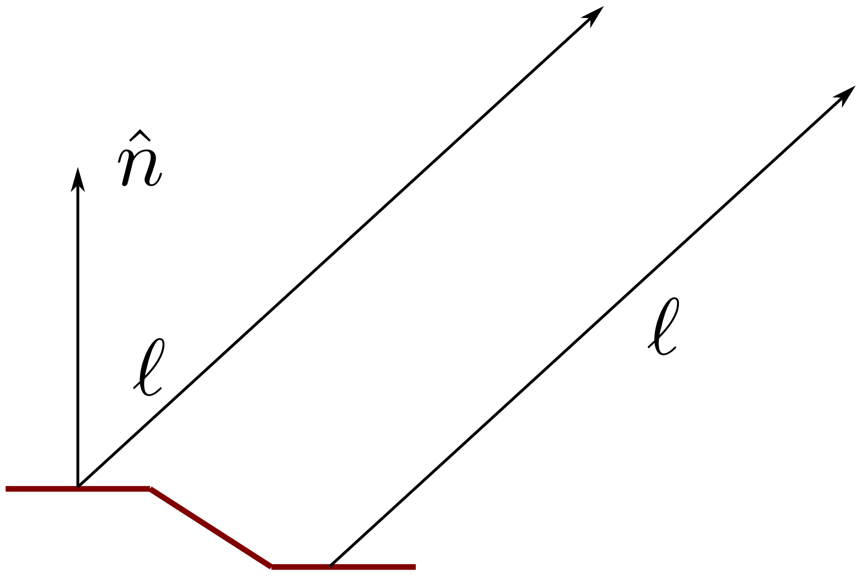
LIGHTS

Point Source at Infinity

For Lambertian surface, reflected radiance in all directions, and therefore observed intensity:



$$I = \rho\langle \hat{n}, \ell \rangle$$



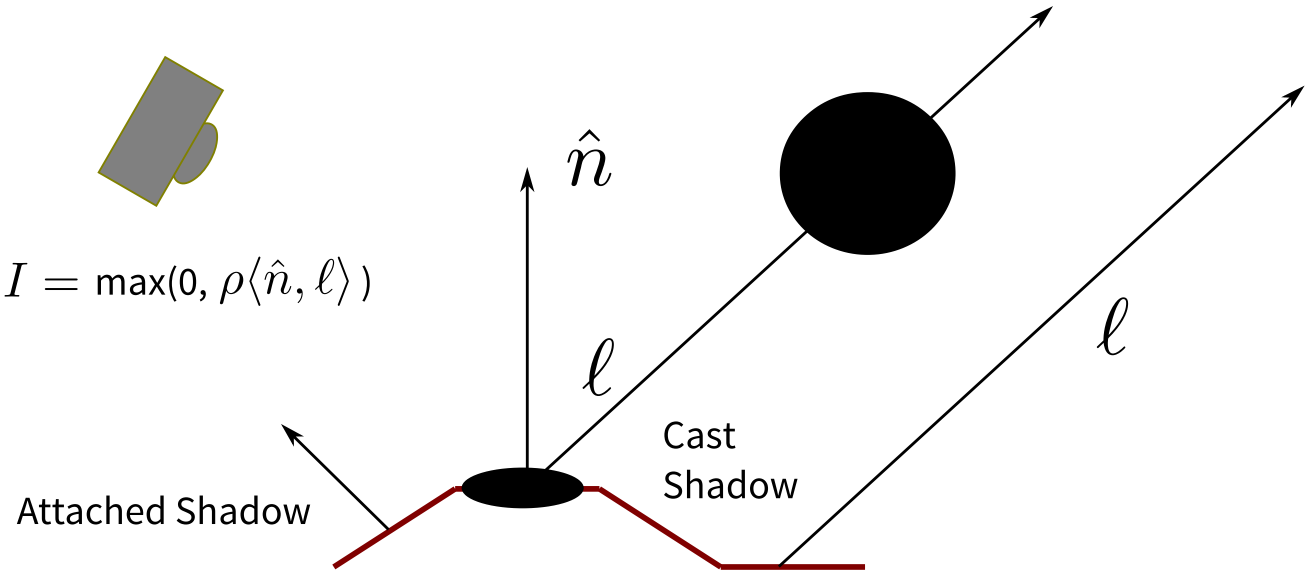
Lighting Direction / Vector is the same everywhere

Good approximation when distance to light source >>> size of the scene.

LIGHTS

Point Source at Infinity

Shadows

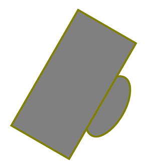


$$I = \text{max}(0, \rho\langle \hat{n}, \ell \rangle\,)$$

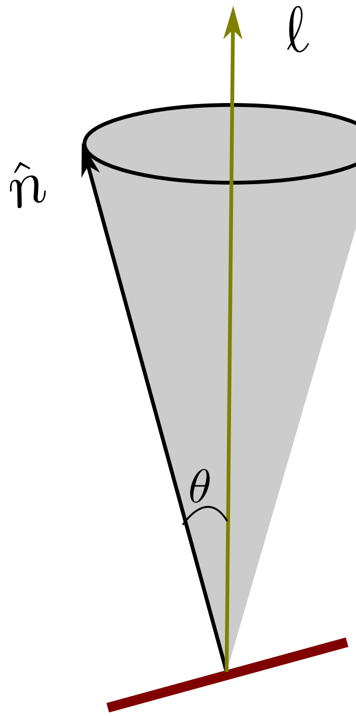
No soft-shadows / penumbra for point light sources.

Lighting Direction / Vector is the same everywhere that has an un-obstructed view of the point light source at infinity.

PHOTOMETRIC STEREO



$I = \rho\langle\hat{n}, \ell\rangle$



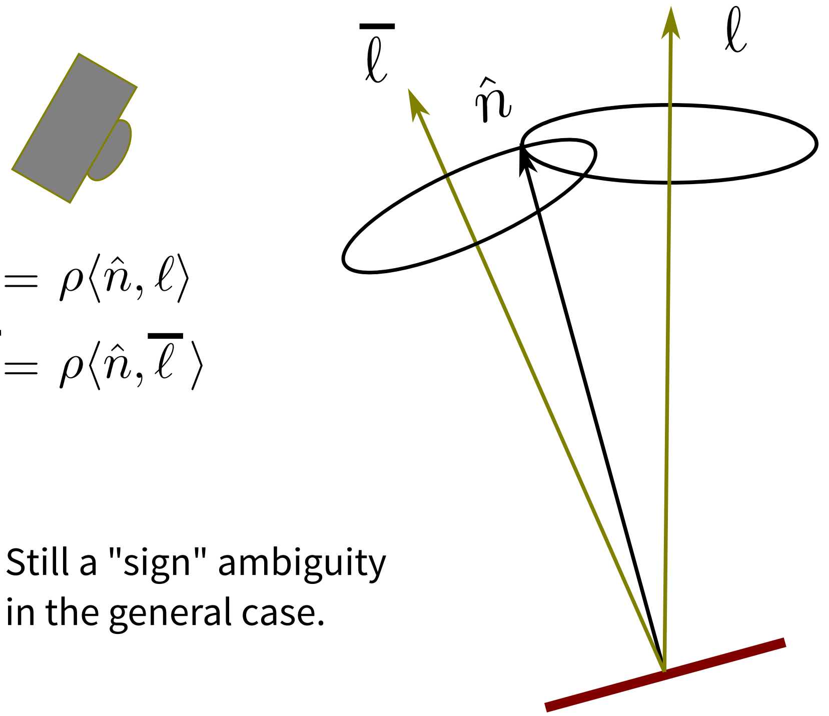
Identifies surface normal upto a cone !

Unless the angle is 0

What does knowing I tell us about $\hat{n}$?

What does knowing I and ρ tell us about $\hat{n}$?

PHOTOMETRIC STEREO



$$I = \rho\langle \hat{n}, \ell \rangle$$

$$\bar{I} = \rho\langle \hat{n}, \bar{\ell} \rangle$$

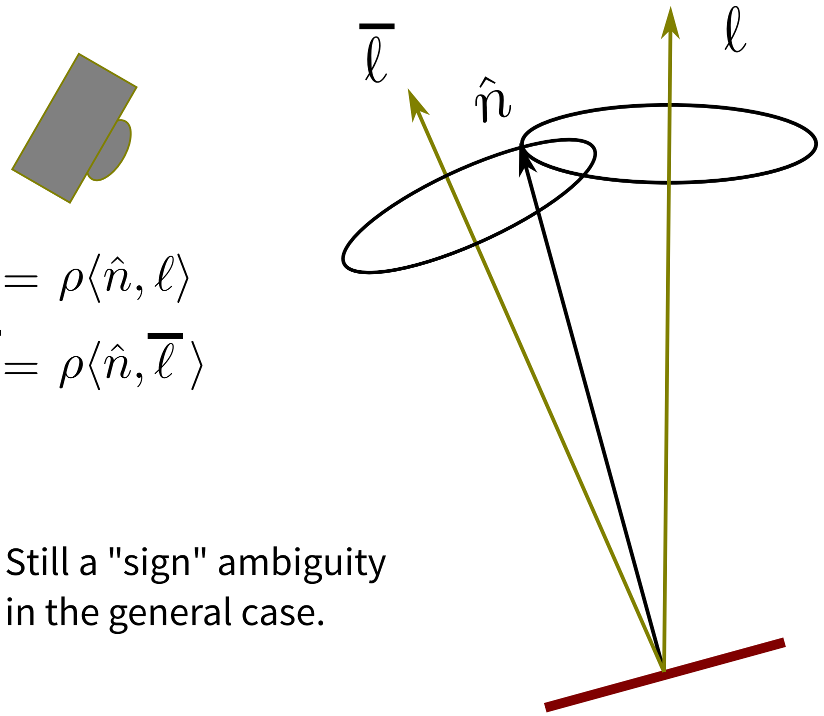
Still a "sign" ambiguity
in the general case.

What does knowing I tell us about $\hat{n}$?

What does knowing I and ρ tell us about $\hat{n}$?

What if you took a second image with a different light direction ?

PHOTOMETRIC STEREO



$$I = \rho\langle \hat{n}, \ell \rangle$$
$$\overline{I} = \rho\langle \hat{n}, \overline{\ell} \rangle$$

Still a "sign" ambiguity
in the general case.

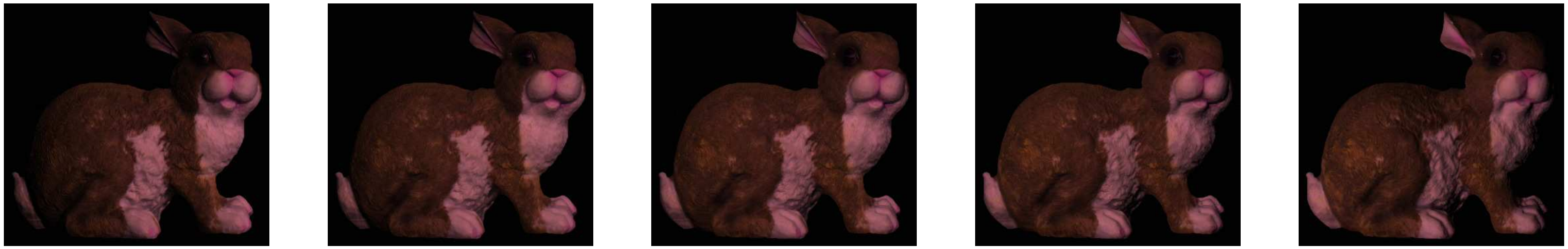
Three measurements, with three linearly independent light vectors, such that the surface isn't in shadow, will uniquely determine normal and albedo.

What does knowing I tell us about $\hat{n}$?

What does knowing I and ρ tell us about $\hat{n}$?

What if you took a second image with a different light direction ?

PHOTOMETRIC STEREO



For each point, let the set of intensities $\{I_i\}$ be observed under lights $\{\ell_i\}$.

Ignore color, assume I_i is scalar (convert the images to grayscale / R+G+B).

$$I_i = \rho \ \langle \hat{n}, \ell_i \rangle = \rho \ \ell_i^T \ \hat{n} = \ell_i^T \ (\rho \ \hat{n}) = \ell_i^T \ \ n$$

Three observations of I_i with different, linearly independent, ℓ_i will give us n.
Three linear equations in three variables.

Given n, we can factor into ρ and $\hat{n}$: ρ is length of $\|n\|$, and $\hat{n} = n/\|n\|$.

PHOTOMETRIC STEREO

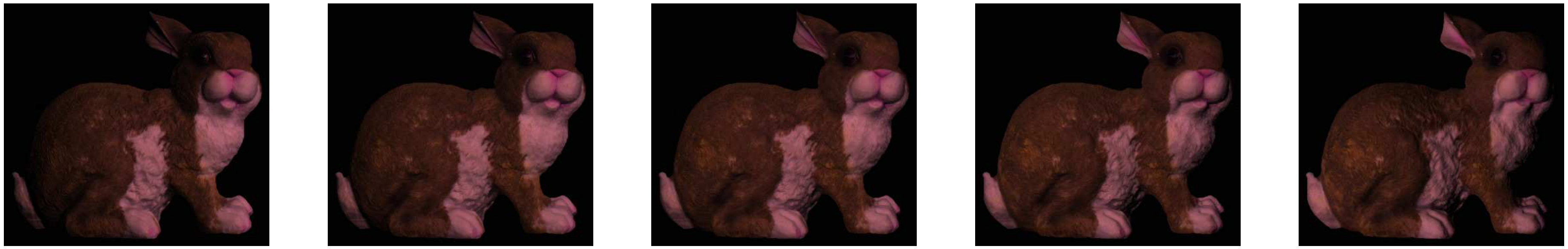
But using only three images is unstable: there will be noise, etc. We solve in the "least squares" sense.

Given K images under K different lights, for each pixel:

$$\begin{bmatrix} \ell_1^T \\ \ell_2^T \\ \ell_3^T \\ \vdots \\ \ell_K^T \end{bmatrix} n = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ \vdots \\ I_K \end{bmatrix} \Rightarrow L\, n = I$$

where L is a $K \times 3$ matrix, I is a $K \times 1$ vector, and n is a three-vector.

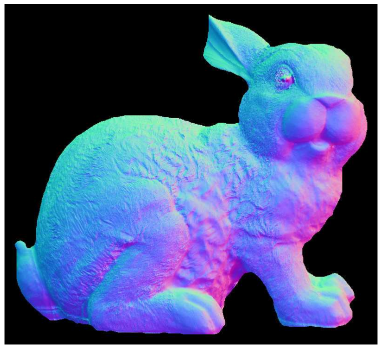
PHOTOMETRIC STEREO



$$n = \arg\min_n \|L\,n - I\|^2 = \arg\min_n n^T\,(L^T L)\,n - 2(L^T\,I)^T\,n + I^T I$$

Take gradient, set to 0: $(L^T L)\,n = (L^T\,I)$

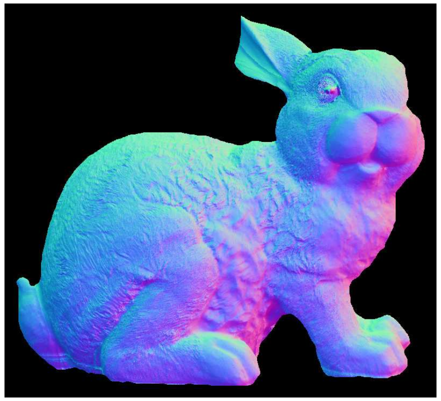
This is actually a 3×3 equation. Solve using `np.lingalg.solve` to use Cholesky.

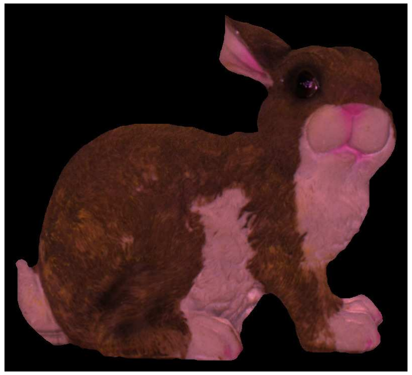


Do this for every pixel in the image.

One of the standard visualizations for normals.
Take normals as a 3 channel image, with n_x, n_y, n_z mapped to R,G,B with [-1,1], mapped to [0,1]

PHOTOMETRIC STEREO





Color Albedos

$$\rho_R = \arg\min_{\rho} \sum_k \left(I_{k:R} - \rho \quad \ell_k^T \hat{n}\right)^2$$

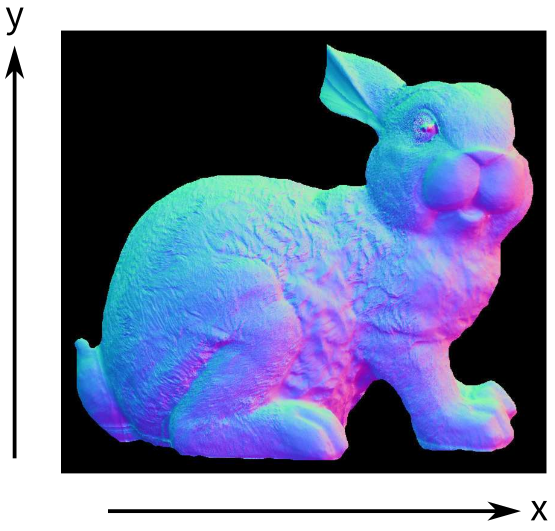
PHOTOMETRIC STEREO

Some Practical Issues:

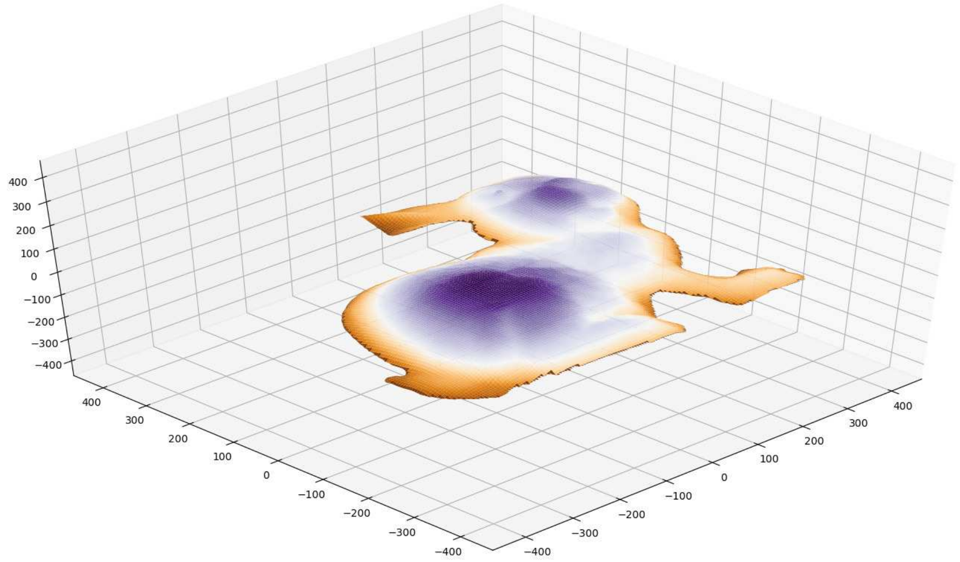
- Even though we assume light at infinity, works well in practice for just far away lights.
- Calibrate light vector ℓ by looking at an image of some known shape and albedo (typically a matte sphere)
- Technically only works for Lambertian objects. But often, can make objects Lambertian with polarizers.
- Also, estimated normals are typically well-defined for a 'valid' set of pixels in the image.
- You'll create / be given a "mask" of these valid pixels.

NORMALS TO DEPTH

Surface Normals



Depth Z(x,y)



$$\frac{\partial Z}{\partial x} = -\frac{\hat{n}_x}{\hat{n}_z}, \frac{\partial Z}{\partial y} = -\frac{\hat{n}_y}{\hat{n}_z}$$

We're going to ignore perspective projection for now.

Assume pixel location (x,y) maps to (x,y,Z(x,y)).

Replace derivatives with finite difference approximations.

NORMALS TO DEPTH

Want to go from a normals "image" to a depth "image"

$$g_x[n] = -\frac{\hat{n}_x[n]}{\hat{n}_z[n]}$$

$$g_y[n] = -\frac{\hat{n}_y[n]}{\hat{n}_z[n]}$$

$$g_x[n] = (Z * \underbrace{\boxed{0.5, 0, -0.5}}^{f_x})[n]$$

$$g_y[n] = \left(Z * \underbrace{\boxed{\begin{bmatrix} -0.50 \\ 0 \\ 0.5 \end{bmatrix}}}_{f_y}\right)[n]$$

De-convolution problem !

n within brackets refers to pixel location.
(too many n's)

NORMALS TO DEPTH

$$Z = \arg\min_{Z} \|g_x - f_x * Z\|^2 + \|g_y - f_y * Z\|^2 + \lambda R(Z)$$

We'll use $R(Z)$ as:

$$R(Z) = \sum_{n} (Z * f_r)[n]^2 \quad \text{for} \quad f_r = \begin{array}{|c|c|c|}\hline -1/9 & -1/9 & -1/9 \\ \hline -1/9 & 8/9 & -1/9 \\ \hline -1/9 & -1/9 & -1/9 \\ \hline \end{array}$$

NORMALS TO DEPTH

$$Z = \arg\min_{Z} \|g_x - f_x * Z\|^2 + \|g_y - f_y * Z\|^2 + \lambda R(Z)$$

Version 1: Do it in the Fourier Domain (called Frankot-Chellappa)

- Assume that in the masked out regions, $g_x = g_y = 0$.

$$\mathcal{F}(Z)[u,v] = \frac{\bar{F}_x[u,v]G_x[u,v] + \bar{F}_y[u,v]G_y[u,v]}{|F_x[u,v]|^2 + |F_y[u,v]|^2 + \lambda|F_r[u,v]|^2}$$

$\mathcal{F}(Z)$ is FT of depth map, G_x is FT of g_x, F_x is (circular padded) FT of f_x, and so on.

- In general, should add some very small number (e.g., 10^{-12}) to denominator for stability.

- In particular, what is the denominator for $[u,v] = [0,0]$?

- Both numerator and denominator are 0, because normals tell us nothing about average depth / offset.
- Explicitly set $\mathcal{F}(Z)[0,0]$ to 0.

NORMALS TO DEPTH

$$Z = \arg\min_{Z} \sum_{n} w[n](g_x[n] - (f_x * Z)[n])^2 + \sum_{n} w[n](g_y - (f_y * Z)[n])^2 + \lambda R(Z)$$

Version 2: Use conjugate gradient.

- Allows us to use different weights for different pixels.
  - Set $w[n]$ to 0 for masked out pixels.
  - Set $w[n]$ to $(\hat{n}_z[n])^2$ everywhere else.
    - Accounts for the fact that we got gradients by dividing by $\hat{n}_z$.
    - Smaller values will be noisier.

NORMALS TO DEPTH

$$Z = \arg\min_Z \sum_n w[n](g_x[n] - (f_x * Z)[n])^2 + \sum_n w[n](g_y - (f_y * Z)[n])^2 + \lambda R(Z)$$

$$Z = \arg\min_Z Z^T QZ - 2Z^T\, b + c$$

- Begin with all zeroes guess Z_0 for Z
- $k = 0, r_0 \leftarrow b - Q\, Z_0, \quad p_0 \leftarrow r_0$
- Repeat (for say a fixed number of iterations)
  - $\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T Q p_k}$
  - $Z_{k+1} \leftarrow Z_k + \alpha_k p_k$
  - $r_{k+1} \leftarrow r_k - \alpha_k Q p_k$
  - $\beta_k \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
  - $p_{k+1} \leftarrow r_{k+1} + \beta_k p_k$
  - $k = k + 1$

NORMALS TO DEPTH

$$Z = \arg\min_Z \sum_n w[n](g_x[n] - (f_x * Z)[n])^2 + \sum_n w[n](g_y - (f_y * Z)[n])^2 + \lambda R(Z)$$

$$Z = \arg\min_Z Z^T QZ - 2Z^T\, b + c$$

We need to figure out:

- What is b (should be same shape as image)
- How do compute $Q\,p$ for a given p (where p is same shape as image)

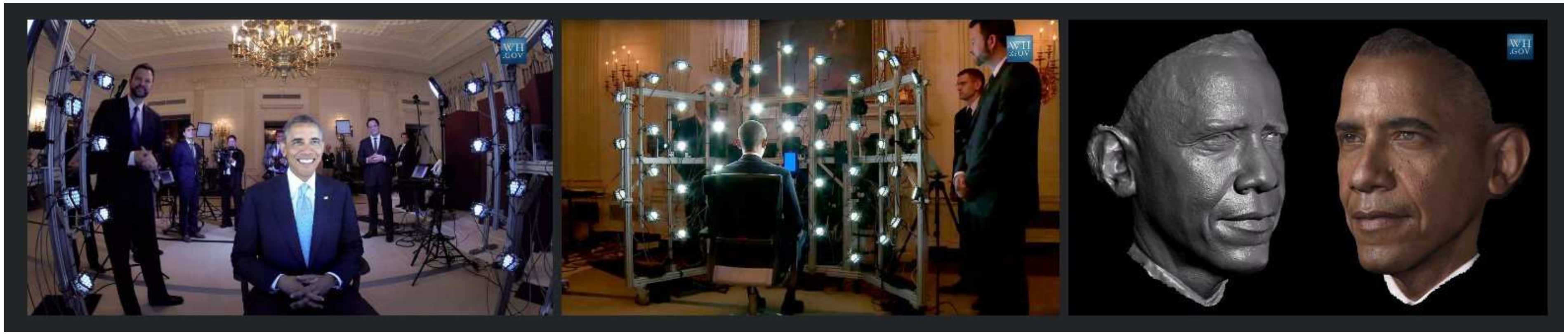
$$Q\,p = ((p * f_x) \times w) * \bar{f}_x + ((p * f_y) \times w) * \bar{f}_y + \lambda((p * f_r) * \bar{f}_r)$$

$\bar{f}$ mean the flipped versions of f.
$\times$ means element-wise product.
$*$ can be same convolutions with zero padding.

$$b = (g_x \times w) * \bar{f}_x + (g_y \times w) * \bar{f}_y$$

- Remember, $p^T Qp$ is just $\langle p, Qp \rangle$. So compute Qp, take element-wise product with p, and sum across all pixels.

PHOTOMETRIC STEREO



NEXT CLASS

- Wrap-up of Photometric / Radiometric Vision
  - Quick overview of variants of photometric stereo
  - General Lighting Environments
  - General Materials
  - RGB Photometric Stereo
  - Unknown Illumination
  - Shape from Shading
- On to Geometric Vision